Public Comments on NIST Special Publication (SP) 800-89 Recommendation for Obtaining Assurances for Digital Signature Applications

Entrust Comments on Draft SP 800-89 Assurances for Digital Signatures
Due 4/28/2006 Draft of 4/18/2006

Entrust is available to discuss any of these comments with NIST.

Please contact Don Johnson at djohnson@cygnacom.com and Robert Zuccherato at robert.zuccherato@entrust.com.

Overall comment: Many of our comments try to point out that while the specified methods in this draft do achieve the assurances desired, there are other possibilities that also work and can be more appropriate. Some of our more important comments below seek to allow these other possibilities via modular approaches that also build on existing methods.

1. Section 3.1 (typo) "Assurance message" should be before any of the "Assurance of…" items.

2. Section 3.2: (typo) "TTA" should be after "TSP" in the alphabetical list.

3. Section 4 and Section 5.1 says "The owner **shall** know which method(s) of assurance were used in order to determine that the provided assurance is sufficient and appropriate to meet the owner's requirements." This does not take into account the scenario where there is an agent (for example, the System Administrator) for the owner either designated by the owner or by an authority over the owner. This agent would make sure the needed assurance was sufficient, thereby relieving the owner of this responsibility.

4. Section 4.1: We note that the requirement for Explicit Domain Parameter Validation for DSA requires the use of the *domain_parameter_seed* and *counter* produced by the domain parameter generation routine. However, these values are not currently included in the domain parameters commonly used for DSA. For example, the Dss-Parms structure in RFC 3279 (Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile) does not include these values. Thus, practical implementation of this requirement will likely need to be coordinated with new parameter structures being defined and used.

5. Section 4.1, items 4 and 5: (A) The generator g being partially valid or verifiably valid is not distinguished sufficiently as both return the final outcome of VALID. Suggest that 2 VALID states be returned; "VALID-g fully validated" and "VALID-g partially validated" to distinguish these 2 different conditions. At least then if one wants g to be generated canonically in a random fashion, one will know what to check.

(B) The point should be made in the text that the generally applicable solution is to generate g so that it is able to be validated. It would seem best for NIST to only allow non-canonical generation of g for backwards compatibility.

6. Section 4.2: Note that a third party George may be a trusted party for one user Ann and may be an untrusted party for another user Bill, depending on the trust domains. That is, whether a third party is a trusted party is decided by the user (or an agent acting for a group of users), not the third party. This needs to be brought out in the text.

7. Section 4.2.1: (A) In an X.509 PKI environment, domain parameter assurance obtained from trust in a TTP will require an indication within the certificate that the TTP (usually a CA) has generated or validated the domain parameters. We note that such an indication (likely in the form of a certificate extension) is not currently defined. Thus, practical implementation of this requirement will likely need to be coordinated with new certificate extensions being defined and used.

   (B) We further note that in order to interoperate with individuals from other domains, whose TTP may not be trusted in the current domain, client software may need to implement Explicit Domain Parameter Validation (or be able to obtain validation from a service). Thus, it can be redundant to require a CA to obtain assurance of this validation as well (as required in the first paragraph of Section 4). The cost and interoperability issues associated with implementing new certificate extensions to support this requirement may not be worth the benefit when clients will have to implement the validation step anyway.

   (C) Therefore, we strongly suggest a modular approach as follows: Identify that it is the relying party that must have this assurance, which can be obtained by (1) The relying party itself doing domain parameter validation, (2) A modular service provided by a TTP (possibly a CA, an RA, or a standalone TTP server) doing domain parameter validation in a client-server model (ala OCSP), or (3) An indication in the certificate provided by a CA (which will need the definition of a certificate extension). This way the needed assurance can be provided to the relying party in a way that is most appropriate for the application. Note that in this three-fold proposal there is no requirement that a CA must have assurance of domain parameter validity and transfer this assurance via a trust relationship with the relying party; this may be the case, but it is not required.

8. Section 5.2: (A) In an X.509 PKI environment, assurance of public key validity obtained from trust in a TTP will require an indication within the certificate that the TTP (usually a CA) has validated or (re)generated the public key. We note that such an indication (likely in the form of a certificate extension) is not currently defined. Thus, practical implementation of this requirement will likely need to be coordinated with new certificate extensions being defined and used.

(B) We further note that in order to interoperate with individuals from other domains, whose TTP may not be trusted in the current domain, client software may need to implement Public Key Validation (or be able to obtain validation from a service). Thus, it can be redundant to require a CA to perform this validation as well (as required in the second paragraph of Section 5). The cost and interoperability issues associated with implementing new certificate extensions to support this requirement may not be worth the benefit when clients will have to implement the validation step anyway.

(C) Therefore, we strongly suggest a modular approach as follows: Identify that it is the relying party that must have this assurance, which can be obtained by (1) The relying party itself doing public key validation, (2) A modular service provided by a TTP (possibly a CA, an RA, or a standalone TTP server) doing public key validation in a client-server model (ala OCSP), or (3) An indication in the certificate provided by a CA (which will need the definition of a certificate extension). This way the needed assurance can be provided to the relying party in a way that is most appropriate for the application. Note that in this three-fold proposal there is no requirement that a CA must have assurance of public key validity and transfer this assurance via a trust relationship with the relying party; this may be the case, but it is not required.

9. Section 5.3 Item 4 says: “This method is not preferred, since the TTP will know the private key and must be trusted not to masquerade as the owner.” In some cases, this method may actually be preferred, so this text is too negative in tone. For example, if a company distributes public key pairs to its employees for intracompany signatures, it can easily be a preferred solution for the company to generate key pairs for all employees, as then a high quality key pair generator can be known to be used for all generation. This is just one example. It is true that this validation method is not preferred IF TTP key pair generation is not preferred, but in some cases TTP key pair generation may be preferred and when this is the case, this validation method may also be preferred.

10. Section 5.3.3 RSA Partial PKV: (A) We note that ALL methods rely on a quality random number generator and that this requirement cannot be validated for any specific instance of a public key pair. Thus, it is impossible for a third party to validate that all requirements for any key pair are met, regardless of algorithm. This point should be added to the text for all methods.

    (B) The ANSI X9F1 workgroup is discussing methods for full RSA key pair validation. Including a method for full RSA key pair validation would likely address possible concerns on not having a full RSA PKV method and specifying only plausibility tests for an RSA public key (considered by itself). Yet there is no discussion of this possibility as to whether it would be allowed or not. 11. Section 6 says “Therefore, cases 2 and 3 are less desirable than case 1, where, by design, only the owner knows the private key.” This is an overstatement; there can be scenarios where case 2 or case 3 is actually the preferred method. An example of case 2 is where a company provides key pairs its employees for intra-company use and therefore can ensure that a high-quality key pair generator is used to generate all key pairs. An example of case 2 or case 3 is where the TTP needs to serve as a backup repository for the signing key, in scenarios where the owner may lose it but this loss is not acceptable due to requirements for continuous processing. These are just examples. It is true that as a general statement and all else being equal, case 1 is more desirable than either case 2 or 3; but there are examples where “all else” is not equal.

12. Section 6 says “In the case of DSA, the private key is denoted as *x*; for ECDSA, the private key is *d*. No explicit assurance of possession is required for the DSA and ECDSA per-message secret number *k* (which may be considered as a key). For RSA, the private key is the pair (*n*, *d*) or an equivalent representation.” This is simplistic, can be misleading and lead to false intuitions.

    In the case of DSA, the private key is denoted as the private value *x* coupled with the public domain parameters; for ECDSA, the private key is denoted as the private value *d* coupled with the public domain parameters. For RSA, the private key is denoted as the private value *d* coupled with the public modulus *n* or an equivalent representation. In each of these 3 cases, the public value portion of the private key defines the group structure and the private value defines the secret inside that group structure. This information should be added to the text as it will both be more correct and help unify the reader’s conceptualization of a private key.

13. Section 6.1 Item 3 says “Implementation errors are most likely to exist when the implementation has not been validated by a NIST Approved testing laboratory.”

    We think better wording would be something like “Validation by a NIST Approved testing laboratory can help provide assurance that implementation errors do not exist.”

    An error can exist or not and testing may or may not detect the error if it exists. Doing validation testing increases the assurance there are no implementation errors, but whether there are errors or not is a different (albeit related) issue relating to programmer skill, possible compiler errors, etc.

14. Sections 6.1 and 6.2: The (a, b, c) models presented here have not been discussed in a public forum as far as we know. These requirements do not seem to us to be well thought out. We strongly suggest that public discussion be done (e.g., in ANSI X9F1) before incorporation in a NIST document. At least the following needs to be addressed:

    (A) Are the values a, b, and c required? Letting a user specify these values can get very complex, but producing values that are reasonable for all applications appears difficult.

(B) Are the HIGH, MEDIUM and LOW classifications meaningful? It is not clear how to determine the appropriate classification and what, in practice, it means.

(C) Section 6.1 and 6.2 (at the end of each) says "After degrading in level, the process of explicitly obtaining/providing assurance of private key possession **shall** be repeated if a higher level of assurance is required." Should not this reassurance be done BEFORE degrading in level, to allow for providing continuous operation?

However, see item 21 for our primary working proposal in this area.

15. Section 6.3: In an X.509 PKI environment, assurance of private key possession obtained from trust in a TTP will require an indication within the certificate that the TTP (usually a CA) has obtained the necessary assurance. We note that such an indication (likely in the form of a certificate extension) is not currently defined. Thus, practical implementation of this requirement will likely need to be coordinated with new certificate extensions being defined and used.

    We also note that if our recommendation from item 21 is adopted, then clients can assume that a default assurance of private key possession level had been obtained by the CA and no additional certificate extensions would be required. This is the approach that has been successfully taken with the PKIX environment and would seem to have less additional costs or interoperability issues.

16. Section 6.3: It is not clear why item 1 is not the only method specified. Even if either of the 2 items in item 2 was done, why would one NOT want to do item 1, after all, this provides assurance that the entire signature process works, not just the key generation process. That is, doing item 1 would seem to be a normal and expected part of certifying a signing key anyway, so why not make things simple and just make this the only way? Before "unleashing" an owner with his signing key to sign things and send there all throughout a domain it seems prudent to make sure the entire signing process had worked successfully at least once. Sending out a million signatures before verifying even one can mean a lot of wasted bandwidth and instruction cycles if something in the signing process has a mistake. There is also a simplicity argument, why be more complex than needed? We believe Section 6.3 item 2 and Section 6.3.2 should be removed.

17 Section 6.3: If item 2 is not going to be removed per item 16, then it should be split into items 2 and 3, as the concepts are different.

18. Section 6.3.1.1: We are concerned that NIST has decided to define a new message for obtaining assurance of private key possession. The industry standard certificate management protocols (RFC 4210 - Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP), RFC 4211 – Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF), RFC 2797 - Certificate Management Messages over CMS) do not support this new message format. These protocols support obtaining assurance of private key possession (called proof-of-

possession in the RFCs) using a slightly different procedure. Either these standards will now need to be modified to accommodate this new assurance message, or CAs will need to implement an additional message to obtain an assurance that they already have. Either approach will be costly. Thus, we would recommend that the proof-of-possession methods included in the current industry standards be considered as possible method of obtaining this assurance. Practical implementation of a requirement for a new message format will likely need to be coordinated with new or modified protocols being defined and used. We also note that if our recommendation from item 21 is adopted, then existing protocols can be used in the majority of environments. This is the approach that has been successfully taken with the PKIX environment and would seem to have less additional costs or interoperability issues

19. Section 6.3.1.1: Points 1 and 2 require that the signatory's identity and the intended verifier's identity must be included in the private key possession assurance message. This could be problematic however in situations where one or both identities are not known. For example, upon initial registration within a PKI an end entity may not know his/her identity within the PKI or the identity of the CA until after obtaining his/her certificate. Of course, the CA should not produce a certificate until after all the needed assurances have been obtained. To deal with this problem, RFCs 2797, 4210 and 4211 include a pseudo-identity value derived from a secret shared by the end entity and the CA, instead of the identities, within the message. We suggest that this option be provided for this message as well.

20. Section 6.3.1.3: The discussion about HIGH, MEDIUM and LOW assurance of possession seems misguided in our opinion as it mixes concepts and oversimplifies the realities all the while being stated as SHOULDs and not as SHALLs, all of which contribute to possible confusion. The overall concept of degradation of assurance over time is useful. There seem to be 2 key concepts, the quality of the time stamp indicating the start of assurance and the amount of time until degradation of that assurance. This would indicate that a more straightforward model would state (A) the quality of the start-of-assurance time stamp and (B) whether that assurance is considered current or stale. For example, states could be TTA-current, TTA-stale, TTP-current, TTP-stale, owner-current, owner-stale, verifier-current, verifier-stale, etc. A key could have multiple states; for example TTA-state and verifier-current would allow processing to continue, if that is what the verifier assessed as reasonable. However, even this (potentially) improved method would need many changes to existing processes and it may not meet the needs of some users anyway. See our primary working proposal in item 21.

21. Our primary working proposal to handle assurance of ownership is as follows: Existing methods that show ownership at some point (i.e., past ownership) are deemed sufficient for many users today. For those users that need the increased assurance of current ownership, it seems that an active protocol is needed anyway; knowing that a user owned it a week ago or even just a day ago may not be good enough, as some will want to know if the other party owns it right now. This has the advantage of building on existing protocols and only paying the cost for assurance of

current ownership for those parties that deem this necessary. Therefore our primary proposal is to utilize the existing methods (i.e., the existing PKIX protocols mentioned earlier) to obtain assurance of past ownership (perhaps modified slightly if absolutely needed), and specify a protocol for active users to obtain assurance of current ownership, for example, before signing a high value message. This active protocol and the format that it would take should be discussed in a public forum, perhaps ANSI X9F, before being included in this document. At this point, it is not clear whether the protocol is required, when it would be used or exactly what assurances it should provide. This new protocol could include the use of a TSA when available and appropriate.

22. Section 7 contains the 2 phrases: “A digital signature verifier requires assurance of the claimed signatory’s identity.” and “a verifier may allow cases where the signatory is anonymous.” This needs to be clarified as to why these 2 concepts are not contradictory and can both be true, especially as the latter phrase has no examples to help indicate what is meant.

From: "Wallner, Debbie M" <dmwalln@orion.ncsc.mil>
Date: Thu, 27 Apr 2006 10:53:31 -0400

Thank you for the opportunity to comment on draft Special Publication 800-89, Recommendation for Obtaining Assurances for Digital Signature Applications. The following general comment is provided for your consideration in future revisions of this document.

Debby Wallner
National Information Assurance Research Laboratory, NSA

The document currently reflects an expectation that key pairs are either generated solely by the intended owner, or solely generated by a TTP. As a result, the various assurance strategies ignore, or at worst exclude, the case in which a key pair is generated with assistance from a TTP, but in such a way that the private key is known only by the intended owner. The document needs to be reviewed to ensure that all key possession and validation techniques can be clearly interpreted for that case. By way of example, Section 5.1 very strictly limits the cases of generation to those of owner solely, and TTP solely. On the other hand, Section 6 is already almost general enough in the list of generation techniques for a key pair.

Date: Wed, 26 Apr 2006 09:21:51 -0400
From: "McRea, Holly R" <McReaHR2@state.gov>

Comment Number	Comment Type (G-General, E-Editorial, T-Technical)	Section or Page Number	Comment(Include rationale for comment)	Suggested change
1	E	Section 3.1, pg 6	The content of this list would be easier to read if it appears in tabular form (i.e., with the lines).	Recommend that the definitions and acronyms appear in tables with the appropriate grid lines for easier reading.
2	G	Section 3.1, pg 6 (Approved)	The latter half of this definition does not agree with the equivalent definition in SP 800-56A (draft).	Recommend that such identifications be exact duplicates, and that this definition be revised to match that in SP 800-56A.
3				Recommend that the following be added to the end, "..., or (b) in a document referenced by the FIPS or NIST Recommendation.
4	T	Section 3.1, pg 7 (Owner)	Although this definition contains essentially the same thought as SP 800-56A, it lacks the specificity of that definition.	Recommend this definition be modified to incorporate the differences between static and ephemeral key pairs, and the fact that the owner is not necessarily the entity that generated the key pair.
5	E	Section 3.1, pg 8 (Shall)	This definition lacks full specificity.	Recommend incorporating some / all of the verbiage from the definition in SP 800-56A ("This term is used to indicate a requirement of a Federal Information Processing Standard (FIPS) or a requirement that needs to be fulfilled to claim conformance to this Recommendation. Note that shall may be coupled with not to become shall not.").

6	E	Section 3.1, pg 8 (Should)	This definition lacks full specificity.	Recommend incorporating some / all of the verbiage from the definition in SP 800-56A ("This term is used to indicate an important recommendation. Ignoring the recommendation could result in undesirable results. Note that should may be coupled with not to become should not.").
7	E	Section 3.1, pg 8 (Trusted Third Party)	This definition lacks full specificity.	Recommend incorporating some / all of the verbiage from the definition in SP 800-56A (A third party, such as a CA, that is trusted by its clients to perform certain services. By contrast, the initiator and responder in a scheme are considered to be the first and second parties in a key establishment transaction.).
8	E	Section 4, pg 9 (first paragraph)	The wording in this paragraph is very cumbersome. Although it disseminates all of the necessary information, persons not highly familiar with PKI and Certificate Policies (CPs) are likely to become confused.	Recommend that the paragraph be reworded (perhaps along the same lines as either the FBCA CP or the FCPF).
9	G/T	Section 4, pg 9 (second paragraph)	According to the definitions above, an "entity" may be an individual (person), organization, device or process. However, experience indicates that individual users, even less so than devices and process, will do this on their own. If it is not set-up as an automatic function working in	Recommend rethinking this paragraph to either better specify which "entity" will perform this parameter check and/or require the Certification Authority to conduct it silently.

			the background and without direct user intervention, it "shall not" be accomplished.	
10	T	Section 4, subpara. 3, pg 10	The sunset date for SHA-1 has already been established, if not exactly "codified."	Recommend that the "probable" date for sunsetting SHA-1 be used instead of the vague, "At some time in the future,"
11	G/T	Section 4.2, pg 11 (first & second paragraphs)	There is a third variant that is not addressed here -- that of a hybrid in which many/most domain parameters are generated by a TTP but some entity members of the domain are authorized and do generate their own.	Recommend that this variation be addressed.
12	G/T	Section 5, pg 12	The comments to paragraph 4 above also apply here. "Entities" (persons, organization, device or process) are unlikely to perform this function on their own. Unless it is set-up as an automatic function working in the background and without direct user intervention, it "shall not" be accomplished.	Recommend rethinking this paragraph to either better specify which "entity" will perform this parameter check and/or require the Certification Authority to conduct it silently.

13	G/T	Section 5.1, subparagraph 4, pg 13	While signature keys are always generated by the "client," it is common practice that an RA or LRA actually generates the keys (as on a smart card) on behalf of the user. Having an end user (such as in method 1) generate their own keys under their own security controls is not preferable to having a TTP do it under established security controls. Its also not preferable from a help desk or ease of use perspective for end users to try and do this themselves. This is also not in sync with PIV workflow which does not have the end user generating their key pairs.	Recommend that this provision be reviewed and reworded to account for the realities of the majority of digital signature uses to date. Revise the phrase "the TTP will know the private key" to "the TTP will be in possession of the private key" to more accurately reflect the reality of the situation in most cases. Also recommend that "This method is not preferred, since the TTP will know the private key and must be trusted not to masquerade as the owner" be rethought.
14	T	Section 6, pg 16 (third paragraph P	The issue here appears to be the meaning of the word "know(s)." For example, if the Certificate Authority (i.e., the machine) generates the key pair(s) and certificate(s), which are downloaded to a FIPS 140-2, Level 2 token; and the token is securely transferred to the individual user in a secure manner, then the meaning of	Recommend an initial clarification of TTP to indicate that when a TTP is trusted and authorized for key generation that any potential risks are understood and have been accepted; therefore any scenarios involving malicious activity by a TTP will not be distinguished from malicious activities of an untrusted third party

			the word "know" as applied to the private key is altered.	
15	G/T	Section 6.1, subpara. 1, pg 17	While this may be a valid example, it overlooks a category of non-technical owner actions that can break the binding between the keys and the owner -- and thereby the trust in the use of those keys. For example, a change in the owner's legal identity could invalidate this binding and necessitate the creation of new keys bound to the new identity.	Recommend that the full scope of possibilities be considered and discussed.
16	G	Section 6.1, pg 18 (first paragraph)	Do these levels of "assurance_time" equate in any way to the more commonly known and understood levels of assurance relating to the CAs and the certificates that are issued.	Recommend that such a relationship, if any, be explained in sufficient detail to clarify any/all similarities and /or differences.
17	G/T	Section 6.5.1, pg 27	None of the methods specified for the owner to obtain an acceptable level of assurance of private key possession seem to be well thought out. It is not relevant to the verification of a	Recommend rethinking the real objective of this process and how it can be implemented in a way that is manageable and useful

			signature. This process outcome cannot be communicated to the verifier. Additionally, it is the verifier's responsibility to determine assurance, so it is irrelevant as to whether the owner determines their possession.	
18	G/T	Section 6.5.2, pg 30	"The TTP shall provide both the recorded *assurance_time* and the initial assurance level associated with the assurance of private key possession provided by the owner in a response to a request by any relying party." This does not allow for the current certificate use model where the TTP (CA) is not directly involved with the relying part. In this case, there is no mechanism (and no need) for the relying party to request this info from the CA.	Revise this section to acknowledge current (and accepted) conventions in using certificates to communicate owner possession of private key
19	G	Complete document	If this document's target audience was PKI implementers, enablers, operators, managers, etc, then this document is completely miswritten and purposefully obscured to make the real intent and mechanisms	

			incomprehensible. There is no direct relationship established between the requirements of this document and the known conventions and mechanisms of PKI (e.g. CAs, CRLs, lifetimes, revocations, timestamps, etc)	

Date: Thu, 27 Apr 2006 15:32:09 -0400
From: "Portnoff, Eliot" <Eliot.Portnoff@hq.doe.gov>

Department of Energy
Comments for:
(SP 800-89)
Recommendation for Obtaining Assurances for Digital Signature Applications

28APRIL06

I. Introduction

Thank you for this opportunity to comment on the Recommendation for Obtaining Assurances for Digital Signature Applications (SP 800-89) document. We are providing the following comments below:

II. General Comments

In general, we believe that the document is well written, however the document should be edited for proper defining of all used acronyms (capitalize as necessary), punctuation, and accurate referencing between information within the document itself. Consider the referencing of Sections and Paragraphs, ex. Section 4.0, Paragraph 4.1, Paragraph 4.2 etc. Please see Section IV of these comments for specific editorial remarks. We have no additional comments to provide in respect to technical content; the document appears sound from that aspect.

III. Technical Comments

Section Reference	Comments
NONE	No Technical issues were noted.

IV. Editorial Comments

Section Reference	Comments
Table of Contents	**Format Paragraph 6.3.1.2:** This paragraph's page location reference may need to be changed. It is currently displayed as an orphan from the paragraph's title with the text located on the following page, (pgs. 21, 22).
Add Figure Table	None
Section vs. Paragraph Referencing	Consider referencing text as main sections and paragraphs, ex. Section 4.0, Section 5.0, Paragraph 4.1, Paragraph 4.2, Paragraph 5.1, and Paragraph 5.2. etc.
Definition of Acronyms	Define the following acronyms when they first occur: DSA, RSA, ECDSA, DRGB, FIPS, ANS, CA, and PKCS.
3.2 Acronyms	Add the following: FISMA – Federal Information Security Management, OMB – Office of Management & Budget, CMVP – Cryptographic Module Validation Program, DRGB - ? (Ref. Paragraph 2), PKI – Public Key Infrastructure, ANS - ? (Ref. Paragraph 4.2.2), SHA-1 - ? (Ref. Paragraph 4, subparagraph 3.), and FIPS – Federal Information Processing Standard.
Capitalize Acronyms	**T**rusted **T**hird **P**arty (TTP) pg 5. , 8
Use Acronyms Only	TTA pg. 8, 16, 21, 26, 28, 29, 31; CA, pg. 16
Delete Acronym; Use Definition	TTP pg. 8, TTA pg. 9
Paragraph 2	**Clarify:** Use of the word furtherance: What is being furthered? Is this in respect to the expansion, clarification, or enforcement of statutory responsibilities?
Page 7, (Key)	Add NIST to read: "Examples applicable to this **NIST** Recommendation include:…"
Page 8	Rewrite to read: "verification of the digital signature **and attainment of** the appropriate assurances…."
Paragraph 4.1	Add information (exact reference to) concerning **ANS X9.62** as referenced in paragraph 4.2.2.
	INTENTIONALLY LEFT BLANK
Paragraph 6.5.3	**Format:** The word "**level**" is a widow on the following page.
Page 5	Add comma to text subpara. 4 to read: "….assurance of the validity of the public keys (see Section 5)**,** ……"

Section Reference	**Comments**
Paragraph 6.4	Add comma to text: Paragraph 6.4 text should read: “….using values for a, b, c and d that are known**,** and have been….”
Paragraph 6.5.1	Edit text to read: “An intended owner shall generate **his/her** own key pair using the Approved….”
Paragraph 6.5.1; Page 27, Item 2 c.	Appendix Reference: Reference should be Appendix **A**, not 6.3.1.1
Section 4.0	Add the word “the” to the beginning of the sentence. Acronyms should never be at the beginning of a sentence. Text should read: “**The** DSA and ECDSA depend on the….”
Section 5.0	Add the word “the” to the beginning of the sentence. Acronyms should never be at the beginning of a sentence. Text should read: “**The** CAs shall have assurance of public key validity….”
Paragraph 3.2	Omit periods at the end of each item; this is a list of items
Paragraph 6.5.1, item 1a.	Add the word the in the text to read: “In accordance with the assurance of **the** possession model….”
Paragraph 6.5.1, item 2a.	Add the word the in the text to read: “In accordance with the assurance of **the** possession model….”
Paragraph 6.5.2, items 1a., 2a.,	Add the word the in the text to read: “In accordance with the assurance of **the** possession model….”
Paragraph 6.5.3, items 1a. 2a	Add the word the in the text to read: “In accordance with the assurance of **the** possession model….”
Paragraph 6.5.3, item 2c.	Capitalize the word “the” to read: “**The** TTP shall provide….”